НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»

Факультет електроніки
(повна назва інституту/факультету)

Акустики та акустоелектроніки
(повна назва кафедри)

«До захисту допущено»
Завідувач кафедри
_____ Дідковський В.С.
(підпис) (ініціали, прізвище)
“___”_____20__ р.

Дипломна робота
на здобуття ступеня бакалавра

зі спеціальності (спеціалізації)_____6.050803 Акустотехніка____
(код та назва спеціальності)

на тему:_____Алгоритми розпізнавання усної мови_____

Виконала: студентка __4__ курсу, групи _ДГ-51_
(шифр групи)

_____Дон Маргарита Едуардівна_____ _____
(прізвище, ім’я, по батькові) (підпис)

Керівник_____професор, д.т.н. Продеус А. М._____ _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант _____ _____ _____
(назва розділу) (посада, вчене звання, науковий ступінь, прізвище, ініціали) (підпис)

Рецензент _____ _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ – 2019 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Інститут/факультет _____електроніки_____
(повна назва)

Кафедра _____акустики та акустоелектроніки_____
(повна назва)

Рівень вищої освіти – перший (бакалаврський)

Спеціальність (спеціалізація) ____6.050803 Акустотехніка_____
(код і назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Дідковський В.С.
(підпис) (ініціали, прізвище)
«___»_____20__ р.

ЗАВДАННЯ
на дипломний проект (роботу) студенту
_____Дон Маргариті Едуардівні_____
(прізвище, ім'я, по батькові)

1. Тема проекту (роботи)__Алгоритми розпізнавання усної мови_____
_____
_____
керівник проекту (роботи) _____Продеус А.М., професор, д.т.н._____,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом по університету від «___»_____ 20__ р. №_____

2. Строк подання студентом проекту (роботи) _____

3. Вихідні дані до проекту (роботи) математичні моделі для побудови алгоритмів розпізнавання усної мови, обробка та аналіз отриманих даних по розпізнаванню усної мови, аналіз існуючих середовищ для розпізнавання усної мови, методи практичної реалізації підвищення ефективності автоматичного розпізнання мови, _____

4. Зміст (дипломної роботи) пояснювальної записки (перелік завдань, які потрібно розробити) _____
1. Елементи теорії розпізнавання мови._____
2. Програмний інструментарій._____
3. Експериментальні дослідження шляхів підвищення ефективності автоматичного розпізнавання мови._____
4. Практична реалізація підвищення ефективності автоматичного

розпізнавання мови._____

5. Перелік графічного (ілюстративного) матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо) __презентація____

6. Консультанти розділів проекту (роботи)*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання _____20 вересня 2018 р_____

Календарний план

№ з/п	Назва етапів виконання дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Збір матеріалів для роботи. Аналіз науково-технічної літератури.	20.09.2018 - 21.10.2018	
2	Ознайомлення з існуючим програмним інструментарієм для розпізнавання усної мови, його аналіз.	21.10.18 - 03.12.18	
3	Обробка отриманих даних та обрахунок якості розпізнавання мови.	03.12.18 - 25.02.19	
4	Надання рекомендацій щодо покращення ефективності екрана при його проектуванні. Вибір акустичного центра шуму транспортного потоку.	25.02.19 - 15.05.19	
5	Оформлення пояснювальної записки та презентації.	15.05.2019 - 05.06.2019	

Студент _____ _____М. Е. Дон_____
(підпис) (ініціали, прізвище)

Керівник проекту (роботи) _____ ____А. М. Продеус___
(підпис

* Консультантом не може бути зазначено керівника дипломного проекту (роботи)

РЕФЕРАТ

Алгоритми розпізнавання усної мови // Дипломна робота на здобуття ступеня вищої освіти «бакалавр». Дон М. Е. Національний технічний університет України «Київський політехнічний інститут ім. Ігоря Сікорського», факультет електроніки, кафедра акустики та акустоелектроніки, група ДГ-51. – К.:НТУУ «КПІ», 2019. с. – 57, рис. – 14, табл. – 3.

Метою роботи є аналіз існуючих алгоритмів розпізнавання мови та експериментальне дослідження функціонування програм, призначених для розпізнавання усної мови.

В роботі проведено аналіз існуючих математичних моделей для побудови алгоритмів розпізнавання усної мови: приховані марковські моделі. Проведені експериментальні дослідження залежності якості автоматичного розпізнання мови від таких факторів, як наявність фонового шуму та реверберації. Дослідження показали критичний рівень фонових шумів та тривалості реверберації для розпізнавання усної мови. Запропоновано метод отримання «звукового профілю» мікрофона, за допомогою якого можливо «вирівняти» АЧХ сигналу, записаного на цей мікрофон, з метою поліпшення якості звучання. Описано два методи отримання АЧХ мікрофона, між якими можна здійснювати вибір в залежності від доступних умов запису та апаратури.

Ключові слова: розпізнавання мови, приховані марковські моделі, фоновий шум, реверберація, амплітудно-частотна характеристика.

ABSTRACT

Speech recognition algorithms // Thesis for a degree of higher education "Bachelor". Don M. National Technical University of Ukraine "Kiev Polytechnic Institute named after Igor Sikorsky, Faculty of Electronics, Department of acoustics and acoustoelectronics, group DG-51. - K: NTUU "KPI", 2019. p. - 57, fig. - 14, tab. - 3.

The aim of the work is the analysis of existing speech recognition algorithms and the experimental research of the programs designed to recognize oral speech.

The analysis of existing mathematical models for constructing oral speech recognition algorithms (Hidden Markov Models) is carried out. Experimental studies of the dependence of the automatic speech recognition quality on the factors such as the presence of background noise and reverberation have been carried out. Studies have shown the critical level of background noise and the duration of reverb for oral speech recognition. The method of obtaining a "sound profile" of a microphone is proposed, with the help of which it is possible to "align" the frequency response of the signal recorded on this microphone in order to improve the sound quality. Two methods of reception of the microphone amplitude-frequency characteristics are described, among which it is possible to make choices depending on available recording conditions and equipment.

Key words: speech recognition, hidden Markov models, background noise, reverb, amplitude-frequency characteristic.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

ЕОМ – електронно-обчислювальна машина.

ПММ – прихована марковська модель.

HTK – Hidden Markov Model Toolkit (Інструментарій ПММ).

АЧХ – амплітудно-частотна характеристика.

1. ВСТУП

Розпізнавання мови — це здатність машини або програми ідентифікувати слова і фрази в розмовній мові і конвертувати їх у відповідний формат для машинного зчитування.

Побудова системи автоматичного розпізнавання мови є в даний момент актуальним завданням. Такі системи є важливим кроком у поліпшенні взаємодії між людиною і комп'ютером. Особливо ця ідея розвинена в концепції так званих «розумних будинків». Більш того, іноді голосовий інтерфейс є необхідним компоненто, наприклад, коли мова йде про людей з порушеннями опорно-рухового апарату, зорової та слухової систем. Хорошим прикладом такого роду є система синтезу мови, якою користувався всесвітньо відомий фізик-теоретик Стівен Хокінг, що втратив здатність говорити після операції на горлі в 1985 році. На сьогоднішній день розвитку напрямку мовних технологій за більш ніж 50 років досліджень сприяли дослідження безлічі компаній і дослідників.

Системи автоматичного розпізнавання мови набагато складніші систем мовного синтезу. Завданням таких систем є виділення і розпізнавання з потоку звукового сигналу (як мовного, так і не мовного) заздалегідь визначеного набору мовних команд. При цьому система не повинна реагувати на інші ділянки мовного сигналу, включаючи і ті, які містять окремі слова зумовлених команд.

Створення комп'ютерних систем розпізнавання мови пов'язане з безліччю об'єктивних труднощів, що накладають на подібні системи штучного інтелекту ряд обмежень.

Граничні можливості комп'ютера по розпізнаванню мови обмежені насамперед тим, що людина, яку можна взяти за еталон розпізнавальної системи, розпізнає мовне повідомлення з урахуванням сенсу, що міститься в ньому, це не піддається реалізації в комп'ютерній техніці в повній мірі. Комп'ютер принципово не може з необхідною надійністю виправляти

помилки і неоднозначності розпізнавання, використовуючи синтаксичний та семантичний зв'язок слів в реченні. Замість цього в сучасних системах використовується так звана n-програмна модель, при використанні якої ставиться завдання передбачення найбільш ймовірного елемента (наприклад, слова) в послідовності, що містить n-1 попередників.

Крім того, людина використовує найчастіше додаткову, незвукову інформацію. Прикладом тут може служити так зване «читання по губах», якому можуть навчитися глухі люди. Відомо, що в галасливій обстановці людині легше розпізнавати мову, якщо він стежить за губами мовника. Людина сприймає мову об'ємно, що дозволяє їй виконувати придушення шуму і просторове виділення сигналу більш якісно, ніж ЕОМ. Слуховий апарат людини дозволяє їй з точністю до півпростору визначити напрямок на джерело корисного сигналу і відокремити його від інших звукових джерел.

Фонетичні моделі, використані в програмуванні алгоритмів на ЕОМ, не точні, так як не використовують усього різноманіття факторів, в силу відсутності математичної моделі семантики мовного сигналу. Для завдання фонетичних еталонів зазвичай використовують статистичні та евристичні методи, що не дають точного результату і використовують низку припущень. Це призводить до того, що точна модель еталонів звуків і слів повинна включати в себе безліч еталонних елементів (по одному на кожен варіант вимови).

Додатково картина ускладнюється тим, що всі відомі алгоритми розпізнавання мови є в тій чи іншій мірі дикторозалежні. Крім того, індивідуальні характеристики мовця, такі як специфіка вимови, акценти, наголоси, хезитації (мовні коливання, пов'язані зі спонтанністю мови: мовні збої, заминки в промові, коливання у виборі слова або конструкції), найчастіше залишаються неврахованими. Таким чином, після налаштування на голос одного диктора системи розпізнавання дають задовільні результати розпізнавання для цього типу голосу, але гірше працюють на інших голосах.

Надійність розпізнавання мови людиною, навпаки, мало залежить від типу голосу диктора.

Крім того, навіть у випадку розпізнавання мови диктора, не до кінця вирішеною залишається проблема якості вихідних мовних сигналів - дуже часто мовні сигнали сприймаються за допомогою мікрофонів побутового рівня, в несприятливих для розпізнавання мови умовах. На якість розпізнавання суттєво впливають також такі фактори, як обсяг навчальної вибірки і відмінність характеристик технічних каналів, використаних на етапах навчання і розпізнавання.

Все вищесказане призводить до того, що розпізнавання спонтанної мови комп'ютером має обмежену надійність, підвищити яку до рівня розпізнавання мови людиною, ймовірно, не вдасться в майбутньому ні шляхом вдосконалення алгоритмів розпізнавання, ні шляхом збільшення обчислювальних потужностей комп'ютера.

Разом з тим, дуже велике коло завдань не вимагає розпізнавання злитої спонтанної мови. У даній роботі проведено дослідження залежності якості автоматичного розпізнавання від ряду параметрів, а також дослідження можливості зменшення впливу деяких видів перешкод і чинників, що заважають розпізнаванню.

2. ЕЛЕМЕНТИ ТЕОРІЇ РОЗПІЗНАВАННЯ МОВИ

2.1. Приховані марковські моделі

2.1.1. Ланцюг Маркова

Випадковий процес називається марковським процесом або процесом без наслідків, якщо для кожного моменту часу t_0 ймовірність будь-якого стану системи в майбутньому (при $t > t_0$) залежить тільки від її стану в теперішньому часі (при $t = t_0$) і не залежить від того, коли і яким чином система прийшла в такий стан (тобто як процес розвивався в минулому при $t < t_0$).

Марковські випадкові процеси діляться на два типи: процеси з дискретними станами і процеси з неперервними станами.

Марковський процес називається випадковим процесом з дискретними станами, якщо всі можливі стани системи $S_1, S_2, \ldots, S_i, \ldots, S_n$ можна перерахувати, а сам процес складається з того, що час від часу система S миттєво (стрибками), переходить (перескакує) з одного стану в інший.

Випадкові процеси з неперервним станом — це такі процеси, для яких характерний послідовний, плавний перехід з одного стану в інший. Прикладами таких процесів можуть бути: процес зміни электричних параметрів (напруги, сили струму, ємності) будь-якого ланцюга (системи, блоку), процес зміни кількості палива в транспортному засобі та інші.

Аналіз і математичний опис випадкових процесів з дискретними станами зручно почати з побудови графічної схеми процесу. На такій схемі зазвичай прямокутниками зображують всі можливі стани системи, стрілками — можливі переходи зі стану в стан. Графічне зображення всіх можливих станів системи і переходів з одного стану в інші називають графом станів системи (або процесу) або просто графом. Наприклад, на рис. 1 представлений граф, зображаючий систему, у якої можливих станів $n = 6$, можливих переходів (стрілок) $l = 10$; в загальному випадку $n \neq l$.

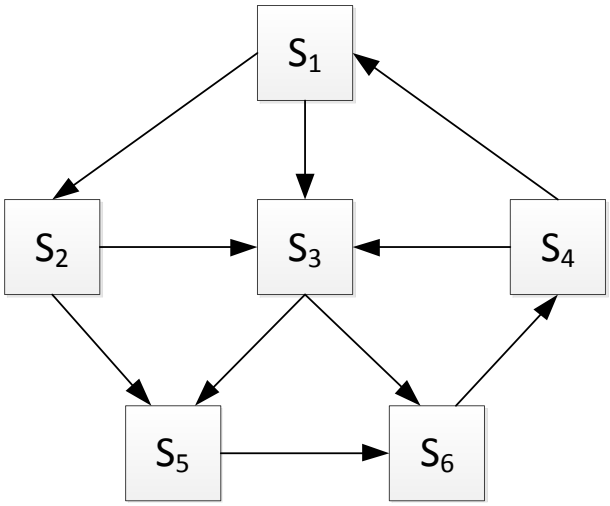


Рисунок 1. Граф станів системи з $n=6, l=10$

Способи математичного опису марковського випадкового процесу, протікаючого в системі з дискретними станами, залежать від того, в які моменти часу — завчасно відомі або випадкові — можуть відбуватися переходи системи з одного стану в інший.

Випадковий процес називається процесом з дискретним часом, якщо переходи системи зі стану в стан можливий тільки в строго зазначені, завчасно фіксовані моменти. В проміжках часу між цими моментами система *S* зберігає свій стан незмінним.

Випадковий процес називається процесом з неперервним часом, якщо перехід системи зі стану в стан можливий в будь-який, завчасно невідомий, випадковий момент часу t.

Марковські випадкові процеси з дискретним часом, протікаючі в системах з дискретними станами, називають ланцюгами Маркова, а марковські випадкові процеси з неперервним часом, протікаючі в системах з дискретними станами, називають неперервними ланцюгами Маркова. [1]

2.1.2. ПРИХОВАНІ МАРКОВСЬКІ МОДЕЛІ В РОЗПІЗНАВАННІ МОВИ

Прихована марковська модель (ПММ) — статистична модель, імітуюча роботу процесу, схожого на марковський процес з невідомими параметрами,

задачою якого ставиться розгадування невідомих параметрів на основі спостерігаємих. ПММ може бути розглянута як найпростіша Байєсівська сітка довіри [2].

Статистичні методи, основані на понятті марковського джерела, і приховані марковські моделі (ПММ) були вперше введені та вивчені ще в кінці 60-х — початку 70-х років. Вперше стали застосовуватись для обробки мови Бейкером (Baker) з CMU, а також Джелінеком (Jelinek) і його колегами з IBM в 1970-х. Ці моделі достатньо змістовні по своїй математичній структурі і, як наслідок, можуть скласти теоретичний фундамент для широкого кола додатків. Окрім того, правильне застосування цих моделей для рішення деяких важливих прикладних задач приводить до дуже гарних результатів.

Тим не менше, широке розповсюдження ПММ в задачі розпізнавання мови отримали відносно нещодавно: початково основи теорії прихованих марковських моделей були опубліковані в журналах для математиків, не дуже популярних серед інженерів, займаючихся розпізнаванням мови; окрім того, опублікована теорія не змістила в собі пояснень про можливості і способах застосування ПММ до різних прикладних областях. Зараз же ці моделі користуються великою популярністю.

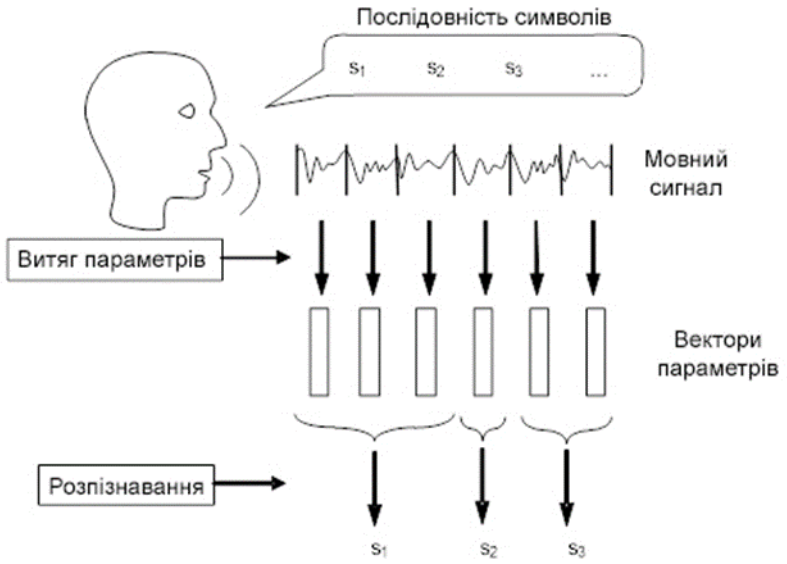


Рисунок 2. Кодировка/декодировка повідомлення

В загальному випадку, при розгляді систем розпізнавання мови припускають, що мовний сигнал — це деяке повідомлення, кодироване за допомогою послідовності одного або декількох символів (наприклад, послідовність звуків — див. рис. 2). Для выконання зворотньої процедури по розпізнаванню послідовності символів, представлених у фрагменті мови, неперервний мовний сигнал спочатку перетворюють в послідовність рівновіддалених в часі векторів дискретних параметрів. Передбачається, що ця послідовність векторів параметрів формує точне представлення форми хвилі мовного сигналу, оскільки на інтервалі часу, охопленим одним вектором (зазвичай приблизно 10 мс), мовний сигнал може рахуватися стаціонарним, що являється розумним приближенням. Зазвичай використовують представлення параметрів, базуючихся на згладжених спектрах або коефіцієнтах лінійного передбачння, а також інші способи представлення, на основі вищезазначених (наприклад, мел-кепстральні коефіцієнти).

Роль системи розпізнавання заключається в тому, що вона має поставити в співвідношенні одна одній послідовності векторів параметрів мови і потрібні послідованості символів. Зробити це дуже важко по двум причинам.

По-перше, перехід від символів до мови не є однозначним: різні символи можуть приводити до появи майже однакових звуків мови. Окрім того, значні зміни мовного сигналу можуть відбутися при зміні диктора, інтонації, обстановки, і т.д. По-друге, неможливо абсолютно точно вказати на млвному сигналі границі між символами. Тому мовний сигнал неможливо трактувати як послідовність з'єднаних незмінних образів.

Другої проблеми, обумовленої незнанням точних границь мовного елементу, можна уникнути, обмежуючись задачою розпізнавання окремих слів. Це означає, що мовний сигнал відповідає єдиному символу у вигляді слова, вибраному з фіксованого словника. Ця спрощеня задача є декілька штучною, тим не менш, вона широко використовується на практиці. Більш того, вона є гарною основою для ознайомлення з базовими ідеями розпізнавання за допомогою ПММ до того як приступити до більш складного випадку злитої промови. Оскільки в даній роботі проводяться эксперименти по распізнаванню ізольванных слів, то в теоретичній частині обмежимося розглядом цього випадку.

Дамо формальне визначення елементам ПММ і пояснимо, як модель генерує спостережувану послідованість. ПММ визначається наступними елементами:

1. N — загальна кількість *станів* в моделі. Не дивлячись на те, що стани в ПММ являються скритими, в багатьох випадках є відповідність між станом моделі і реальним станом процесу. В загальному випадку, перехід в будь-який обраний стан можливий з будь-якого стану всієї системи (в тому числі і саме в себе); з іншої сторони, лише деякі шляхи переходів представляють інтерес в кожній конкретній моделі. Позначимо сукупність станів моделі множиною $S = \{S_1, S_2, \ldots, S_N\}$, а поточний стан в момент часу t як q_t.
2. M, кількість можливих *символів* в спостерігаємій послідовності, розмір *алфавіту спостерігаємої послідовності*. *Алфавіт спостерігаємої*

послідовності позначимо як $W = \{w_1, w_2, \ldots, w_M\}$.

3. *Матриця ймовірностей переходів* (або матриця переходів) $A = \{a_{ij}\}$, де

$$a_{ij} = P\left[q_{t+1} = S_j \mid q_t = S_i \right],\ 1 \le i,\ j \le N \qquad (2.1)$$

тобто це ймовірність того, що система, яка знаходиться в стані S_i, перейде в стан S_j. Якщо для будь-яких двох станів в моделі можливий перехід з одного стану в інший, то $a_{ij} > 0$ для будь-яких i, j. В інших ПММ для деяких i, j ймовірність переходу $a_{ij} = 0$.

4. $B = \{b_j(k)\}$ — розподіл ймовірностей появи символів в *j*-тому стані, де

$$b_j(k) = P\left[w_k \mid q_t = S_j \right],\ 1 \ge j \ge N,\ 1 \ge k \ge M. \qquad (2.2)$$

$b_j(k)$ — ймовірність того, що в момент часу t, система, яка знаходиться в *j*-ому стані (стан S_j), видасть *k*-тий символ (символ w_k) в спостерігаєму послідовність.

5. Розподіл ймовірностей початкового стану $\pi = \{\pi_i\}$, де

$$\pi_i = P\left[q_1 = S_i \right],\ 1 \le i \le N \qquad (2.3)$$

тобто ймовірність того, що S_i — це початковий стан моделі.

Сукупність значень N, M, A, B і π — це прихована марковська модель, яка може згенерувати *спостерігаєму послідовність*

$$O = o_1, o_2, \ldots, o_T \qquad (2.4)$$

(де o_t — один з символів алфавіту W, T — кількість елементів в спостерігаємій послідовності).

ПММ будує спостерігаєму послідовність по наступному алгоритму:

1. Обираємо початковий стан $q_1 = S_i$ у відповідності до розподілу π.
2. Встановлюємо $t = 1$.
3. Обираємо $o_t = w_k$ у відповідності до розподілу $b_j(k)$ в стані (S_i).
4. Переводимо модель в новий стан $q_{t+1} = S_j$ у відповідності до *матриці переходів* a_{ij} з врахуванням поточного стану S_i.

Встановлюємо час $t = t + 1$; повертаємося до кроку 3, якщо $t < T$; в іншому випадку — закінчуємо виконання [4].

Підводячи підсумок, помітимо, що *повний* опис ПММ складається з двох параметрів моделі (N и M), опису символвв спостерігаємої послідовності і трьох масивів ймовірностей — A, B і π. Таким чином, введемо наступний запис для позначення *достатнього* опису параметрів моделі:

$$\lambda = (A, B, \pi). \tag{2.5}$$

2.1.3. Розпізнавання окремих слів

Представимо вимовлене слово послідовністю векторів або *спостережень* O, визначених як

$$\mathrm{O} = o_1, o_2, o..., o_T, \tag{2.6}$$

де o_t є вектором параметрів мови, спостерігаємий в момент часу t. Проблему розпізнавання окремих слів можна розглядати як результат обчислення

$$\arg\max_i \left\{ P\left(\omega_i \middle| O\right) \right\}, \tag{2.7}$$

де ω_i є i-те слово словника. Умовну ймовірність $P\left(\omega_i \middle| O\right)$ не обчислюють безпосередньо, а звертаються до формули Байеса:

$$P\left(\omega_i \middle| O\right) = \frac{P(O \middle| \omega_i) P\left(\omega_i\right)}{P\left(O\right)}. \tag{2.8}$$

Таким чином, при заданній апріорній ймовірності $P(\omega_i)$, найбільш ймовірне вимовлене слово визначається лише правдоподібністю $P(O|\omega_i)$. Через високу мірність послідовності спостережень O пряме оцінювання спільної умовної ймовірності $P(o_1, o_2, ...|\omega_i)$ на екземплярах вимовлених слов не використовується на практиці. Однак якщо зробити припущення щодо *параметричної моделі генерування слова, такої, як Марковська модель*, оцінювання по даним становиться можливим, оскільки *проблема оцінювання умовних щільностей $P(O|\omega_i)$ замінюється набагато простішою проблемою*

оцінювання параметрів Марковської моделі.

При розпізнавання мови, грунтовному на ПММ, *припускається, що послідовність спостерігаємих векторів мови, відповідних якомусь слову, породжена Марковською моделлю,* як показано на рис. 3. [3] [5]

Помітимо, що послідованість станів представленої моделі має ту ластивість, що зі збільшенням часу індекс стану також збільшується (або залишається незмінним), іншими словами стани переходять одного в інший зліва направо. Це так називаєма ліво-права модель, або модель Бакіса. Використання данної моделі дозволяє зменшити кількість можливих послідовностей станів моделі.

На мал. 3 показаний приклад цього процесу, де модель, яка складається з шести станів, проходить через послідовність станів X=1,2,2,3,4,4,5,6, для того, щоб згенерувати послідованість від o_1 до o_6. Помітимо, що вхідне і вихідне стани ПММ не являються породжуючими. Це зроблено для полегшення конструювання важких моделей.

Сумісна ймовірність того, що O згенерована моделлю M, проходячої через послідовність станів X, розраховується просто як добуток ймовірностей переходу і ймовірностей генерування. Так, для показаної на мал. 3 послідовності станів X

$$P\left(O, X \middle| M\right) = a_{12} b_2\left(o_1\right) a_{22} b_2\left(o_2\right) a_{23} b_3\left(o_3\right) \ldots \qquad (2.9)$$

На практиці, однак, відома тільки послідовність спостереження O, тоді як «породивша» її послідовність станів X прихована (від спостереження). Саме тому таку модель называють Прихованою Марковською Моделлю.

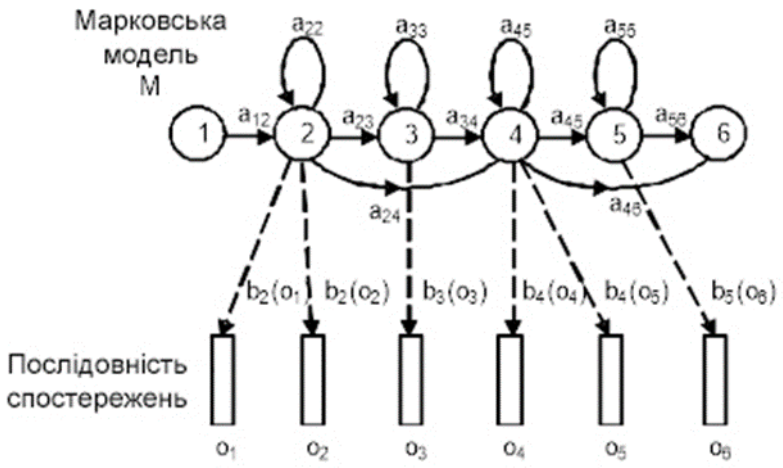


Рисунок 3. Марковська модель генерування послідовності випадкових векторів

Оскільки послідовність X невідома, необхідна правдоподібність обраховується підсумовуванням по всим можливим послідовностям станів, тобто:

$$P\left(O\middle|M\right)=\sum_{X}a_{x(0)x(1)}\prod_{t=1}^{T}b_{x(t)}\left(o_t\right)a_{x(t)x(t+1)}, \tag{2.10}$$

де до $x(0)$ ставиться вимога бути моделлю вхідного стану, а до $x(T+1)$ - бути моделлю вихідного стану.

Як альтернатива виразу (2.10), правдоподібність можете бути вирахувана наближено, шляхам розгляду найбільш ймовірної послідовності станів, тобто:

$$P\left(O\middle|M\right)=\max_{X}\left\{a_{x(0)x(1)}\prod_{t=1}^{T}b_{x(t)}\left(o_t\right)a_{x(t)x(t+1)}\right\}. \tag{2.11}$$

Здійснити прямі обчислення, у відповідності до співвідношень (2.10) і (2.11), не дуже просто, однак існують прості рекурсивні процедури, які дозволяють досить ефективно розрахувати обидві величини. Перед тим, як йти далі, помітимо, що, якщо співвідношення (2.7) може бути обраховано, тоді проблема розпізнавання вирішена. Для заданної множини моделей, відповідних словам, співвідношення (2.7) вирішується з використанням (2.8) і в припущенні, що

$$P\left(O\middle|\omega_i\right) = P\left(O\middle|M_i\right). \qquad (2.12)$$

При цьому припускається, що параметри a_{ij} і $b_j(o_t)$ відомі для кожної моделі M_i. В цьому і складається витонченість і потужність ПММ структури. Для заданої множини прикладів навчання, відповідних конкретній моделі, параметри цієї моделі можна визначити автоматично за допомогою надійної і эфективної рекурентної процедури. Таким чином, за умови, що зібрано достатня кількість представницьких зразків кожного слова, може бути побудована ПММ, яка *неявно моделює всю множину причин мінливості*, властивій реальній мові.

2.1.4. Опис ймовірності породження

Перед тим як детально обговорювати проблему параметричного оцінювання, потрібно прояснити вид розподідень $b_j(o_t)$. В залежності від моделюючих параметрів, розподілення багатомірних щільностей ймовірності можуть бути як неперервними, так і дискретними. Для простоти, будемо припускати, що використовуються неперервні щільності розподілення.

В більшості систем, працюючих з неперервними щільностями, розподілення описуються гаусівськими сумішами щільностей ймовірності. В цьому випадку формула розрахунку $b_j(o_t)$ має вигляд:

$$b_j(o_t) = \sum_{m=1}^{M} c_{jm} N\left(o_t, \mu_{jm}, \Sigma_{jm}\right), \tag{2.13}$$

де M – число змішуваних компонентів, c_{jm} – вага m-го компоненту, а $N(o, \mu, \Sigma)$ – багатомірний Гаусівський розподіл з вектором середнього значення μ і коваріаційною матрицею Σ:

$$N(o, \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^n |\Sigma|}} \exp\left(-\frac{1}{2}(o-\mu)' \Sigma^{-1} (o-\mu)\right) \tag{2.14}$$

де n – розмірність o.

2.1.5. Рекурентне оцінювання Баума-Уелча (Baum-Welch)

Щоб визначити параметри ПММ, спочатку необхідно зробити грубе припущення про те, якими вони могли би бути. Як тільки це зроблено, можна знайти більш точні параметри (в сенсі максимальної правдоподібності) за допомогою так називаємої рекурентної процедури Баума-Уелча.

Можна припустити, що компоненти суміші являються спеціальною формою стану більш низького рівня, в якому ймовірності переходу являються вагами суміші (див. рис. 4).

Таким чином, важливою задачою є оцінювання середніх і дисперсій

ПММ, в якій кожний стан вихідних даних представляється єдиним Гаусівським компонентом:

$$b_j(o_t) = \frac{1}{\sqrt{(2\pi)^n |\Sigma_j|}} \exp\left(-\frac{1}{2}(o_t - \mu_j)' \Sigma_j^{-1} (o_t - \mu_j)\right) \quad (2.15)$$

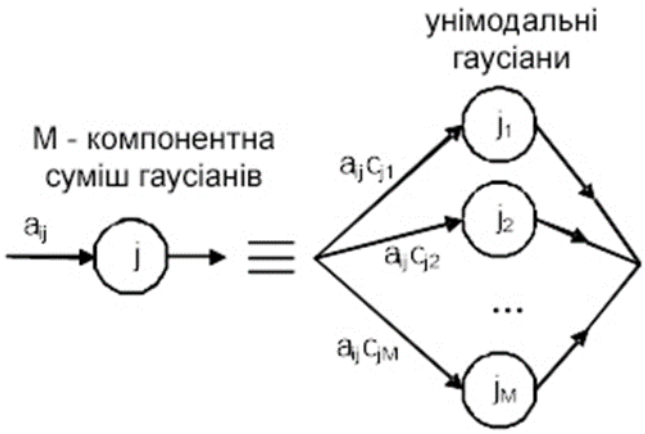


Рисунок 4. Представлення суміші

Якби в ПММ був тільки один стан, оцінити параметр було б легко. Оцінки максимальної правдоподібності величин μ_j і Σ_j можна було б отримати простим опосередковуванням:

$$\hat{\mu}_j = \frac{1}{T}\sum_{t=1}^{T} o_t; \quad (2.16)$$

$$\hat{\Sigma}_j = \frac{1}{T}\sum_{t=1}^{T} (o_t - \mu_j)(o_t - \mu_j)'. \quad (2.17)$$

На практиці, звичайно, існує декілька станів, і неможливо безпосередньо прив’язати вектор спостережень до окремих станів, оскільки базові послідовності станів невідомі. Помітимо, однак, що, якщо б можна було здійснити деяку наближену прив’язку векторів до станів, тоді можна було б використовувати рівняння (2.16) і (2.17) для отримання потрібних початкових значень параметрів. Потім, з використанням описаного нижче алгоритма Вітербі, знаходиться найбільш правдоподібна послідовність станів, вектори спостережень знову прив’язуються до станів, після чого знову використовуються рівняння (2.16) і (2.17) для отримання кращих

початкових значень. Цей процес повторюється до тих пір, поки оцінки перестають змінюватись. Так как повна правдоподібність кожної послідовності спостережень основується на додаванні всіх можливих послідовностей станів, кожний вектор спостереження вносить свій вклад в розрахунки значень параметрів максимальної правдоподібності для кожного стану. Іншими словами, замість того, щоб прив'язувати кожний вектор спостереження до певного стану, як це робилось у вищезазначеном приближенні, кожне спостереження прив'язується до кожного стану, пропорційно ймовірності стану моделі при спостереженні цього вектора. Таким чином, якщо позначити через $L_j(t)$ ймовірність перебування в стані j в момент часу t, приведені вище рівняння (2.16) і (2.17) стають наступними зваженими середніми:

$$\hat{\mu}_j = \frac{\sum_{t=1}^{T} L_j(t) o_t}{\sum_{t=1}^{T} L_j(t)}; \tag{2.18}$$

$$\hat{\Sigma}_j = \frac{\sum_{t=1}^{T} L_j(t)\left(o_t - \mu_j\right)\left(o_t - \mu_j\right)'}{\sum_{t=1}^{T} L_j(t)}. \tag{2.19}$$

де підсумовування в знаменниках забезпечує необхідну нормалізацію. Рівняння (2.18) і (2.19) описують процедуру рекурентного оцінювання Баума-Уелча для средніх і дисперсій ПММ. Аналогічна, але декілька більш важка, процедура може бути отримана для ймовірностей переходу.

Звичайно, щоб застосувати співвідношення (2.18) і (2.19), потрібно розрахувати ймовірність стану $L_j(t)$. Це ефективно робиться з використанням так званого *алгоритма прямого-зворотнього ходу (Forward-Backward algorithm)*. Нехай пряма ймовірність $\alpha_j(t)$ для деякої моделі M з N станами визначена у вигляді:

$$\alpha_j(t) = P\left(o_1, ..., o_t, x(t) = j\right). \tag{2.20}$$

Тобто, $\alpha_j(t)$ - спільна ймовірність спостереження перших t векторів мови для стану j в момент часу t. Ця пряма ймовірність може бути ефективно

розрахована за наступною рекурентною формулою:

$$\alpha_j(t) = \left[\sum_{i=2}^{N-1} \alpha_i(t-1)\alpha_{ij} \right] b_j(o_t). \quad (2.21)$$

Вид цієї рекурентної формули визначається тією обставигою, що ймовірність перебування в стані j в момент t, при спостереженнях o_t, можна вивести шляхом додавання прямих ймовірностей для всіх ймовірних попередніх станах елемента i, зважених ймовірностей переходів a_{ij}. Декілька незвичайні границі обумовлені тим, що стани 1 і N не являються породжуючими. Початкові умови для вищезазначеного рекурентного співвідношення мають вигляд:

$$\alpha_1(1) = 1; \quad (2.22)$$

$$\alpha_j(1) = a_{1j} b_j(o_1). \quad (2.23)$$

для $1<j<N$, а кінцеві умови задаються так:

$$\alpha_N(T) = \sum_{i=2}^{N-1} \alpha_i(T) a_{iN}. \quad (2.24)$$

Відмітимо, що з визначення $\alpha_j(t)$ слідує:

$$P(O|M) = \alpha_N(T). \quad (2.25)$$

Отже, обчислення прямої ймовірності дозволяє отримати повну правдоподібність $P(O|M)$. Зворотня ймовірність $\beta_i(t)$ визначається наступним чином:

$$\beta_j(t) = P(o_{t+1},...,o_T \mid x(t) = j, M). \quad (2.26)$$

Як і у випадку прямої ймовірності, ця зворотня ймовірність може бути ефективно вирахувана з використанням наступного рекурентного співвідношення:

$$\beta_i(t) = \sum_{j=2}^{N-1} a_{ij} b_j(o_{t+1}) \beta_j(t+1), \quad (2.27)$$

з початковою умовою:

$$\beta_i(T) = a_{iN}, \quad (2.28)$$

для $1<i<N$, і кінцевою умовою:

$$\beta_1(1) = \sum_{j=2}^{N-1} a_{1j} b_j(o_1) \beta_j(1). \qquad (2.29)$$

Помітимо, що в приведених вище визначеннях пряма ймовірність є спільна ймовірність, тогді як зворотня ймовірність є умовна ймовірність. Це декілька асиметричне визначення є навмисним, оскільки дозволяє визначити ймовірність надходження в стані як добуток цих двох ймовірностей. За визначенням,

$$\alpha_j(t)\beta_j(t) = P\left(O, x(t) = j \mid M\right), \qquad (2.30)$$

звідси

$$L(t) = P\left(x(t) = j \mid M\right) = \frac{P\left(O, x(t) = j \mid M\right)}{P(O \mid M)} = \frac{1}{P}\alpha_j(t)\beta_j(t), \qquad (2.31)$$

де $P = P(O \mid M)$.

Тепер відомя вся інформація, необхідна для здійснення рекурентного оцінювання параметрів ПММ з використанням алгоритма Баума-Уелча.

Кроки цього алгоритма можна представити наступним чином:

1. Для кожного рекурентно оцінюваного векторного/матричного параметру, варто виділити місце в пам’яті для організації додавання в чисельнику і знаменнику виразів (2.18) і (2.19). Ці місця пам’яті представляють собою накопичуючі суматори.
2. Вираховують прямі і зворотні ймовірності для всих станів j і моментів часу t.
3. Для кожного стану j і часу t оновлюють вміст накопичуючих суматорів, використовуючи ймовірність $L_j(t)$ і поточний вектор спостереження o_t.
4. Кінцеві значення наклпичувального суматора використовують для обрахування нових значень параметрів.
5. Якщо значення $P = P(O \mid M)$ для данної ітерації не вище такого для попередньої ітерації, тоді відбувається зупинка, в іншому випадку слід

повторити вищезазначені кроки, використовуючи нові рекурентно оцінені значення параметрів.

З всього вищесказанного слідує, що параметри ПММ рекурентно оцінюються по єдиній послідовності спостереження, тобто по єдиному екземпляру вимовленого слова. На практиці ж, для отримання хороших оцінок параметра, необхідно багато екземплярів одного і того ж слова. Тим не менш, використання багаторазових послідовностей спостереження не призводить до ускладнення алгоритма: приведені вище кроки 2 і 3 повторюються для кожної нової навчальної послідовності.

2.1.6 Розпізнавання і декодування Вітербі

В попередньому розділі описані основні ідеї рекурентної оцінки параметрів ПММ з використанням алгоритма Баума-Уелча. При цьому було відмічено, що ефективний рекурсивний алгоритм обчислення прямої ймовірності дозволяє заодно вирахувати і повну ймовірність $P(O|M)$. Таким чином, цей алгоритм може використовуватися для знахождення моделі, яка максимізує значення $P(O|M_i)$ і, відповідно, може використовуватися для розпізнавання.

На практиці, однак, зручніше здійснювати розпізнавання, основуючись на максимізації правдоподібності послідовності стану, оскільки це легко узагальнити на випадок злитої мови, що невможливо при використанні повної ймовірності. Цю правдоподібність обраховують, використовуючи, в сутності, той самий алгоритм що і при обрахувані прямої ймовірності, з тою лише різницею, що підсумовування замінюється пошуком максимума. Припустимо, що $\varphi_j(t)$, для даної моделі M, представляє максимальну правдоподібність спостереження послідовності векторів мови від o_1 до o_t і пребування в стані j в момент часу t. Цю часткову правдоподібність можна

ефективно вирахувати з використанням наступного рекурентного співвідношення:

$$\phi_j(t) = \max_i \left\{ \phi_i(t-1) a_{ij} \right\} b_j(o_t), \tag{2.32}$$

де

$$\phi_1(1) = 1; \tag{2.33}$$

$$\phi_j(1) = a_{1j} b_j(o_1), \tag{2.34}$$

для $1 < j < N$. Тоді максимальна правдоподібність $P(O|M)$ має вигляд:

$$\phi_N(T) = \max_i \left\{ \phi_i(T) a_{iN} \right\}. \tag{2.35}$$

Пряме обчислення правдоподібності (2.32) веде до втрати значущих розрядів, тому замість цього обчислюють логарифм правдоподібності. При цьому замість рівняння (2.32) отримуємо:

$$\psi_j(t) = \max_i \left\{ \psi_i(t-1) + \log\left(a_{ij}\right) \right\} + \log\left(b_j\left(o_t\right)\right) \tag{2.36}$$

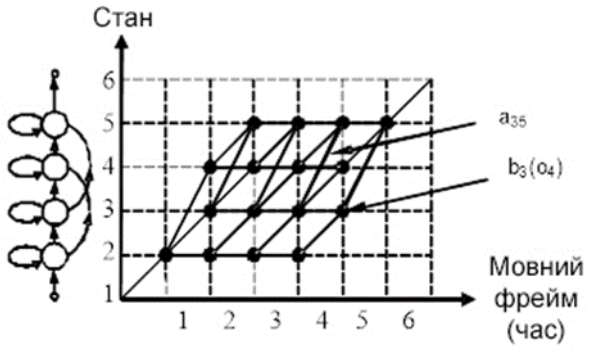


Рисунок 5. Алгоритм Вітербі для розпізнавання ізольованих слів

Це рекурентне співвідношення є основою так званого алгоритма Вітербі. Як показано на рис. 5, цей алгоритм можна представити як виявлення кращого шляху через матрицю, де вертикальний вимір представляє стан ПММ, а горизонтальний вимір представляє фрейми мови (тобто час). Кожна велика точка на картинці представляє логарифм ймовірності спостереження даного фрейму в даний момент часу, а кожний відрізок між точками відповідає логарифму ймовірності переходу. Логарифм

ймовірності будь-якого шляху обраховується простим додаванням логарифмів ймовірностей переходів і логарифмів вихідних ймовірностей вздовж даного шляху. Шляхи йдуть зліва направо, колонка за колонкой. В момент часу *t*, кожний шлях $\psi_i(t\text{-}1)$ відомий для всих станів *i*, тому вираз (2.36) можна використовувати для обчислень $\psi_j(t)$, подовжуючи шляхи на один такт часу.

2.2. Параметризація мовних сигналів

Для розпізнавання мови, заснованого на ПММ, необхідно отримати набір векторів спостереження. В якості векторів спостереження мають бути обрані такі параметри, по яким можна надійно відрізнити один звук мови від іншого. Форма мовного сигналу одного і того ж звуку може суттєво змінюватися, тому відліки мовного сигналу безпосередньо не використовуються. Частіше за все використовують різні методи спектрального аналізу, засновані на перетворенні Фур'є, або на основі лінійного передбачення [3].

2.2.1. Аналіз на основі лінійного передбачення

Лінійне передбачення (LP) є одним з найбільш ефективних методів параметризації мовних сигналів. Цей метод стає домінуючим при оцінюванні основних параметрів мовного сигналу, таких як, наприклад, період основного тону, форманта, спектр, функція площі мовного тракту. Важливість методу обумовлена високою точнісю отриманих оцінок і відносною простою обчислень. Основний принцип методу лінійного передбачення складається в тому, що поточний відлік мовного сигналу можна апроксимувати лінійною комбінацією попередніх відліків

$$s(n) = -\sum_{i=1}^{N}[a_i + s(n-i)] + e(n), \quad (2.37)$$

де N – кількість коефіцієнтів моделі, $e(n)$ — помилка передбачення.

Коефіцієнти передбачення при цьому визначаються однозначно мінімізацією середнього квадрату $e(n)$ — різності між відліками мового сигналу та їх передбаченними значеннями (на кінцевому інтервалі).

Основні положення методу лінійного передбачення добре узгоджуються з моделлю мовоутворення, де мовний сигнал можна представити у вигляді сигналу на виході лінійної системи зі змінними в часі параметрами, збуджувана квазіперіодичними імпульсами (в межах вокалізованого сегменту) або випадковим шумом (на невокалізованому сегменті). Метод лінійного передбачення дозволяє точно і надійно оцінити параметри цієї лінійної системи зі змінними коефіцієнтами. Передавальна функція такої системи:

$$H(z) = \frac{1}{\sum_{i=0}^{p} a_i z^{-i}} \quad (2.38)$$

де p - число полюсів і $a_0 \equiv 1$. Стосовно до мовних сигналів існують наступні методи обчислення параметрів a_i (часто рівноцінні): коваріаційний, автокореляційний, сходового фільтру, зворотньої фільтрації, оцінки спектру, максимальної правдоподібності і скалярного добутку. [3]

2.2.2. Мел-кепстральні коефіцієнти

Окрім коефіцієнтів лінійного передбачення часто для розпізнавання мови використовують коефіцієнти кепстра відрізка мовного сигналу. Кепстр представляє собою Фур'є перетворення від логарифма спектра сигнала.

Такий «спектр спектра» дозволяє отримати характеристики мовного сигналу, які минимально залежать від конкретної реалізації вимовленого

слова. [6]

Назва “кепстр” (“cepstrum”) отримано шляхом реверсії перших чотирьох букв слова спектр (“spectrum”). Вперше був визначений в 1963 Богертом (Bogert) та іншими. Навідмінну від класичного визначення кепстра, в літературі по обробці мови кепстр визначається як зворотнє перетворення Фур’є від логарифму спектра потужності сигналу. В математичному вигляді:

$$C_s = \mathrm{F}^{-1}\left\{\lg\left|\mathrm{F}\{s\}\right|\right\}, \tag{2.39}$$

де F – перетворення Фур'є, F^{-1} – зворотнє перетворення Фур'є, c – кепстр, s – початковий сигнал. Навідмінну від комплексного кепстра, визначений в (2.39) кепстр не містить інформіцію про фазу сигналу. Змінна, від якої залежить кепстр, має розмірність часу, однак, це не той самий час, що у сигнала. Щоб підкреслити, що ця змінна в своєму роді частота для кепстра, іноді вживають назву quefrency (отримано від frequency).

Операція обчислення кепстра відноситься до класу гомоморфної обробки сигналу. Гомоморфні системи – це клас нелінійних систем, які підкорюються загальному принципу суперпозиції. Лінійні системи – це частковий випадок гомоморфної системи. В обробці мови гомоморфні системи мають наступну властивість:

$$D\left[\left[x_1(n)\right]^{\alpha}\cdot\left[x_2(n)\right]^{\beta}\right] = \alpha D\left[x_1(n)\right] + \beta\left[x_2(n)\right]. \tag{2.40}$$

Цей тип суперпозиції відноситься до операцій множення і піднесення до степеню. Такою узагальненою власивістю суперпозиції володіє функція логарифму.

Гомоморфні системи корисні в обробці мови, тому що вони надають метод для розділення форми збуджуючого сигналу і імпульсній перехідній характеристиці голосового тракту. Для розпізнання мови цей підхід интересен з точки зору моделювання характеристик голосового тракту. Розділення двох компонент можна представити як процес деконволюції (зворотня згортка), і може бути описаний наступним чином:

$$s(n) = g(n) \otimes v(n), \tag{2.41}$$

де $g(n)$ – збуджуючий сигнал, $v(n)$ – імпульсна перехідна характеристика голосового тракту, а $\otimes$ - операція згортки. В частотному вигляді згортка представляється як:

$$S(f) = G(f) \cdot V(f). \tag{2.42}$$

Якщо провести операцію комплексного логарифмування над обома частинами, то отримаємо:

$$\mathrm{Log}\big(S(f)\big) = \mathrm{Log}\big(G(f) \cdot V(f)\big) = \mathrm{Log}\big(G(f)\big) + \mathrm{Log}\big(V(f)\big). \tag{2.43}$$

З цього моменту, в логарифмічному сенсі, збудження і характеристика голосового тракту представляють собою адитивну суміш. Зворотнє перетворення Фур'є цієї суміші і буде шуканий кепстр сигналу. Кепстр буде представляти собою суміш імпульсної характеристики і сигналу збудження, які при необхідності можна розділити методами лінійної фільтрації. Інформація про голосовий тракт буде зосереджена в основному в області малих часів кепстра, в той час як в області великих часів заключена інформація про сигнал збудженння.

Замість обчислення Фур'є перетворення сигналу на практиці частіше за все користуються швидким перетворенням Фур'є або використовують гребінку фільтрів. Окрім того, кепстр сигналу також можна отримати з коефіцієнтів лінійного передбачення. Для цього використовується проста рекурсивна формула:

$$c_n = -a_n - \tfrac{1}{n}\sum_{i=1}^{n-1}(n-i)a_i c_{n-i} \tag{2.44}$$

Тут порядок кепстра не обов'язково рівний кількості коефіцієнтів LPC.

Відомо, що людське вухо має різну частотну роздільну здатність на різному діапазоні частот, іншими словами, сприймаєма людським слухом висота звуку залежить від його частоти нелінійним чином. Існує думка, що таке нелінійне перетворення підвищує розбірливість мови. В системах розпізнавання мови для отримання аналогічного перетворення

використовують гребінку фільтрів різної ширини. Одна з таких нелінійних шкал, апроксимуюча шкалу частот людського слуху, називається Мел-частотною. Вона визначається як:

$$\text{Mel}(f) = 1127 \ln\left(1 + \frac{f}{700}\right). \tag{2.45}$$

Приведемо алгоритм отримання мел-кепстральних частотних коефіцієнтів:

1. Вихідний мовний сигнал запишемо в дискретному вигляді як

$$x[n],\ 0 \le n < N. \tag{2.46}$$

2. Примінемо до мовного сигналу перетворення Фур'є

$$X_a[k] = \sum_{n=0}^{N-1} x[n] e^{\frac{-2\pi i}{N} kn},\ 0 \le k < N. \tag{2.47}$$

3. Складаємо гребінку трикутних фільтрів

$$H_m = \begin{cases} 0, & k < f[m-1]; \\ \dfrac{(k - f[m-1])}{(f[m] - f[m-1])}, & f[m-1] \le k < f[m]; \\ \dfrac{(f[m+1] - k)}{(f[m+1] - f[m])}, & f[m] \le k \le f[m+1]; \\ 0, & k > f[m+1]. \end{cases} \tag{2.48}$$

Приклад отриманої гребінки фільтрів для випадку $M = 12$ показаний на рисунку 6. [3]

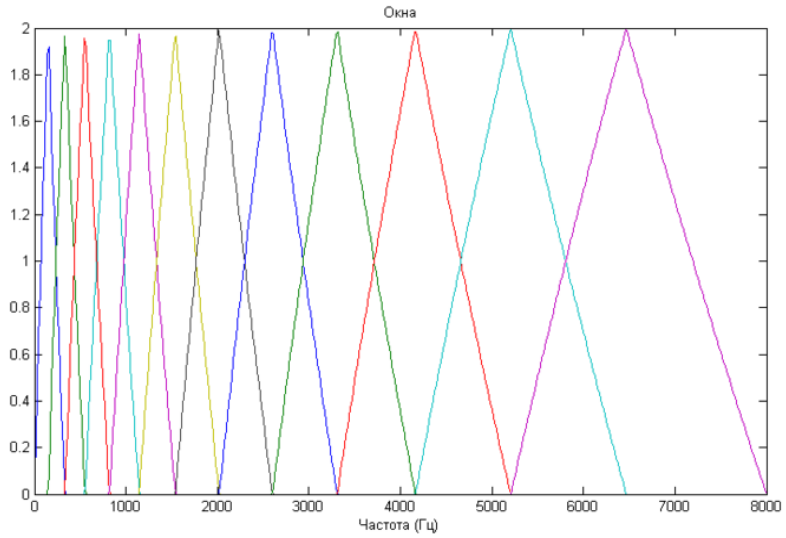


Рисунок 6. Гребінка фільтрів мел-частотної шкали ($M = 12$)

4. Для якої частоти $f[m]$ отримуємо з рівності

$$f[m] = \left(\frac{N}{F_s}\right) B^{-1}\left(B(f_1) + m\frac{B(f_h) - B(f_1)}{M+1}\right). \quad (2.49)$$

5. $B(b)$ — перетворення значення частоти в мел-шкалу, відповідно,

$$B^{-1}(b) = 700\left(e^{b/1125} - 1\right). \quad (2.50)$$

6. Обчислюємо енергію для кожного вікна

$$S[m] = \ln\left(\sum_{k=0}^{N-1} \left|X_a[k]^2\right| H_m[k]\right),\ 0 \le m < M. \quad (2.51)$$

7. Застосовуємо дискретне косинусне перетворення (перетворення Фур'є нема сенсу використовувати, так як обчислення відбуваються над дійсними числами)

$$c[n] = \sum_{m=0}^{M-1} S[m]\cos\left(\pi n \frac{m+0.5}{M}\right),\ 0 \le n < M. \quad (2.52)$$

Значення, отримані за допомогою виразу (2.52), є шуканими мел-кепстральними коефіцієнтами. [6]

Зазвичай трикутні фільтри покривають весь частотний діапазон від нуля впритул до частоти Найквиста. Однак, обмеження полоси пропускання часто корисно, щоб відкинути небажані частоти або уникнути установки

границь фільтрів в частотних діапазонах, в яких немає корисної енергії сигналу [5].

2.2.3. Дельта-коефіцієнти

Ефективність системи розпізнавання мови може бути суттєво збільшена, якщо додатково використовувати похідну по часу від основних статичних параметрів. Ці коефіцієнти названі дельта коефіцієнтами першого порядку, другого порядку (прискорення) і третього порядку. Обчислюють коефіцієнти дельта за допомогою наступної формули:

$$d_t = \frac{\sum_{\theta=1}^{\Theta} \theta(c_{t+\theta} - c_{t-\theta})}{2\sum_{\theta=1}^{\Theta} \theta^2}. \quad (2.53)$$

Коефіцієнти другого і третього порядку обраховують по тій самій формулі, але вже від дельта коефіцієнтів попереднього порядку [3].

3. ПРОГРАМНИЙ ІНСТРУМЕНТАРІЙ

Для того, щоб безпосередньо відтворити та проаналізувати процес розпізнання мови, ми розглянемо можливості двох платформ, а саме – HTK та Python.

3.1 HTK

HTK - інструментарій для роботи з ПММ - навчання, розпізнавання, тестування і т.п. (Англійська абревіатура HTK означає Hidden Markov Model Toolkit, тобто «інструментарій на базі ПММ»). Інструментарій HTK універсальний і може використовуватися для моделювання будь-яких процесів і сигналів, грунтуючись на ПММ. Однак основне його призначення - побудова заснованих на ПММ інструментальних засобів обробки мови, зокрема, систем розпізнавання мови. Перша версія HTK була розроблена в 1989 р Стівом Янгом (Steve Young) працюючи в групі Speech Vision and Robotics Group Кембриджського університету. Ця версія складалася з декількох модулів, написаних на C, і інструментів які використовувалися для досліджень в області розпізнавання мови. Вона була слабо документована і використовувалася тільки вищезгаданою дослідницькою групою в Кембриджі. На даний момент HTK має досить хорошу документацію, вільний для поширення і використання в області досліджень розпізнавання мови. HTK може бути використаний для створення програм розпізнавання мови, але ці програми заборонено використовувати в комерційних цілях. Однак моделі, побудовані з використанням даного інструментарію, можуть бути використані в комерційних продуктах. Незважаючи на це, HTK може успішно використовуватися для створення прототипів систем обробки промові перед розробкою комерційних продуктів на базі відомих алгоритмів.

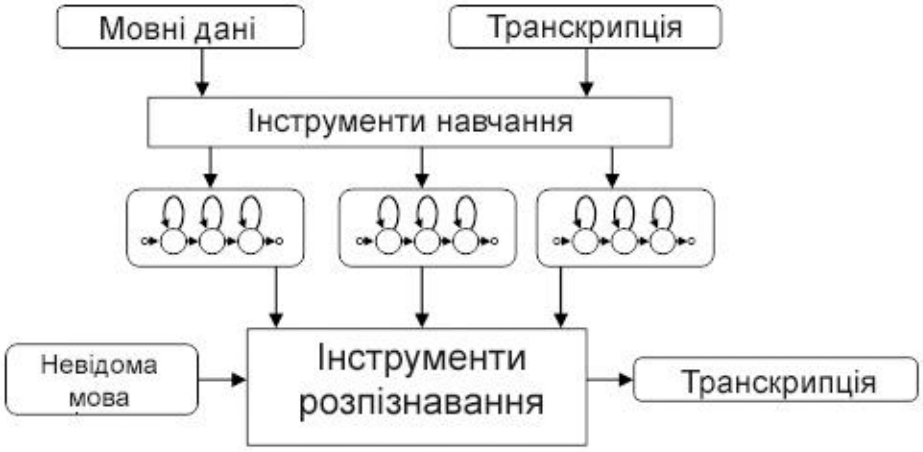


Рисунок 7. Загальна схема роботи з HTK

На малюнку 7 можна виділити дві основні пов'язані стадії обробки. По-перше, навчальні інструментальні засоби HTK застосовуються для оцінки параметрів безлічі ПММ, з використанням навчальних зразків вимови і відповідних їм транскрипцій. По-друге, невідомі зразки проголошення транскрибуються за допомогою засобів розпізнавання HTK. HTK містить безліч інструментів і модулів для роботи з ПММ, які детально описані в керівництві [5].

Інструментарій HTK був написаний на мові програмування C (C ++). Значна частина функціональних можливостей HTK міститься в бібліотечних модулях. Завдяки цим модулям забезпечується однаковість зв'язків інтерфейсів кожного інструменту із зовнішнім світом. Крім того, вони представляють собою центральний ресурс часто використовуваних функцій. На рисунку 8 показана структура програмного забезпечення (ПЗ) типового інструменту HTK, а також показані його інтерфейси вхідних і вихідних даних.

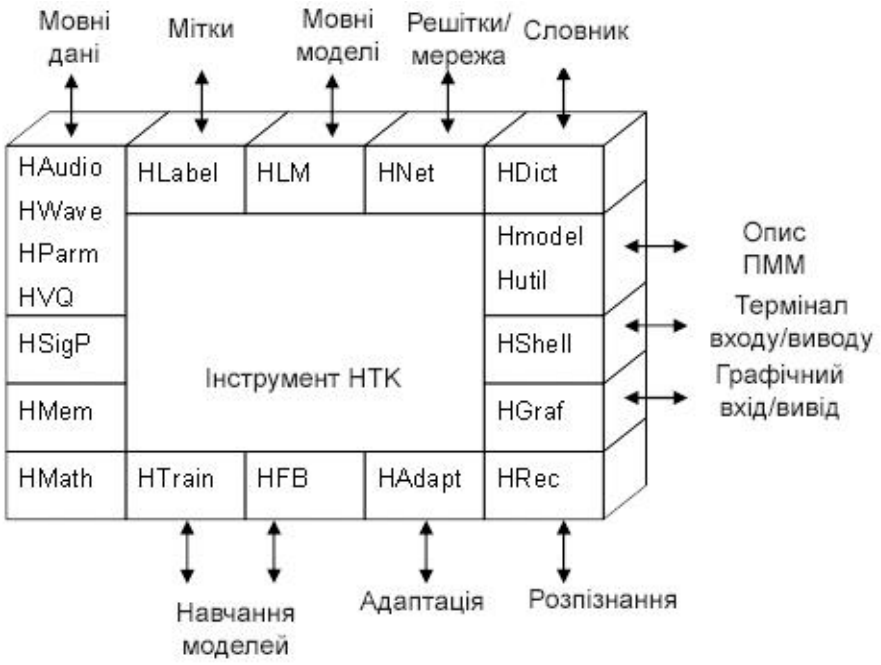


Рисунок 8. Архітектура пакету HTK

Ввід/вивід даних, а також функціонування системи, контролюються бібліотечним модулем HShell, а все управління пам'яттю контролюється HMem. Математична підтримка забезпечується HMath, а необхідні для аналізу мови операції по обробці сигналів виконуються в HSigP. Для кожного типу використовуваних в HTK файлів, є спеціальний модуль інтерфейсу. HLabel забезпечує інтерфейс для файлів міток, HLM - для файлів моделі мови, HNet - для мереж і решіток, HDict - для словників, HVQ - для кодових книг, HModel - для описів ПММ.

Всі мовні вхідні і вихідні дані на рівні форми сигналу проходять через HWave, а на параметризованому рівні - через HParm. HWave і HLable не тільки забезпечують узгоджений інтерфейс, але і підтримують різноманітні формати файлів, дозволяючи імпортувати дані з інших систем. Пряме введення звукової інформації підтримується HAudio, а проста інтерактивна графіка забезпечується HGgraf. HUtil надає ряд стандартних утиліт для маніпулювання з HMM, тоді як HTrain і HBF надають підтримку для різних навчальних інструментів HTK. HAdapt забезпечує підтримку різних адаптаційних інструментів HTK. HRec містить основні функції процедури розпізнавання. HRest дозволяє побудувати HMM ізольованого слова по

набору навчальних зразків, з використанням процедури рекуррентного оцінювання Баума-Уелча. HVite спільно з HNet і HRec реалізує розпізнавання мови, засноване на алгоритмі Вітербі [9].

Головним недоліком HTK є припинення підтримки оновлення програми командою розробників. Через це, робота на деяких версіях сучасних операційних системах є неможливою. Тому експериментальна частина роботи проводилася в наступному розглянутому програмному інструментарію.

3.2 Python. SpeechRecognition library

Python - високорівнева мова програмування загального призначення, орієнтована на підвищення продуктивності розробника і читання коду. На сьогоднішній день, можливості та простота у реалізації мови Python, робить її найбільш поширеною серед базових програм-розпізнавачів мови [8].

На PyPI існує кілька пакетів для розпізнавання мови. Пакет SpeechRecognition розрахована на роботу з декількома популярними мовними API, що робить її дуже гнучкою у застосуванні. Один з них - Google API, який дає змогу розробникам конвертувати аудіо в текст, застосовуючи потужні нейронні мережі в простому API. Google API розпізнає 120 мов для підтримки глобальної бази користувачів. Розробники, за допомогою цього одного інструменту, можуть включити голосові команди та керування, записати аудіо з центрів обробки викликів тощо. API може обробляти потокове передавання в реальному часі або попередньо записаний звук, використовуючи технологію машинного навчання Google[9, 10].

Гнучкість і простота використання пакета SpeechRecognition роблять його відмінним вибором для будь-якого проекту на Python. Проте підтримка кожної функції кожного API, яку вона переносить, не гарантується. Наприклад, для розпізнавання мови в онлайн режимі, потрібно підключити пакет PyAudio, який дасть змогу програмі виявити підключений мікрофон та зчитувати вхідні дані з нього [11].

Вся основна робота відбувається в класі Recognizer завданням якого є, очевидно, розпізнавання мови з джерела звуку. Так як ми вже зупинились на використанні Google API, виходячи з простоти в використанні та задовільної якості роботи, в функціоналі програми ми використовуємо функцію recognize_google() [11].

SpeechRecognition дозволяє легко працювати з аудіофайлами при підключенні класу AudioFile. Цей клас може бути ініціалізований шляхом до аудіо файлу і надає інтерфейс керування для читання та роботи з вмістом файлу. Програмою підтримуються аудіо файли з розширенням WAV(PCM/LPCM формат), AIFF, AIFF-C і FLAC.

Для того, щоб зібрати данні з аудіо файлу, використовують функцію record(). Контекстний менеджер відкриває файл і зчитує його вміст, зберігаючи дані в примірниках AudioFile, з назвою source. Потім функція record () записує дані з усього файлу в примірник AudioData.

4. ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ ШЛЯХІВ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ АВТОМАТИЧНОГО РОЗПІЗНАВАННЯ МОВИ

4.1. Аналіз впливу деяких перешкод і чинників, що впливають на якість розпізнавання

4.1.1. Постановка задачі

Досліджувалася залежність якості автоматичного розпізнавання розмовних речень від наступних факторів:

1) наявність фонового шуму з відношенням сигнал-шум 0, 5, 10 та 15 дБ;

2) наявність реверберaційної завади із часом реверберації: 0,29; 0,6; 0,89; 1,1; 1,4; 2,0; 2,7 с.

Дослідження проводилися експериментальним шляхом, з використанням ПК і мови програмування Python (код програми наведено в Додатку 1).

Для чистоти експерименту було обрано 6 звукових доріжок з різним набором слів та різними дикторами.

4.2. Розпізнавання звукових доріжок з накладеним фоновим шумом

Для перевірки якості програмного інструментарію для розпізнавання мови, симулюємо поширену ситуацію в повсякденному житті, а саме: накладаємо на звукову доріжку з записом мови диктора шуми ресторану та залізничної станції.

Для того, щоб коректно проаналізувати вплив шуму на якість розпізнавання мови, попередньо обрахуємо в середовищі Matlab співвідношення сигнал/шум. Достатньо провести даний обрахунок для шуму з підсиленням 0 дБ, бо цього достатньо для аналізу та порівняння отриманих даних.

Отримані результати обробки доріжок з фоновим шумом ресторану заносимо в табл. 1:

Таблиця 1. Розпізнавання сигналу з накладеним шумом ресторану

№	Чистий сигнал, %	Сигнал з шумом, %	Сигнал з шумом (підсилення шуму 5дБ) , %	Сигнал з шумом (підсилення шуму 10дБ) , %	Сигнал з шумом (підсилення шуму 15дБ) , %	Відношення сигнал/шум
1.	100	100	97	94	85	21.3565
2.	90	72	63	42	24	23.5140
3.	93	83	83	78	76	22.6750
4.	97	94	92	91	50	17.6111
5.	86	79	77	72	64	17.0554
6.	95	89	89	89	76	22.0528

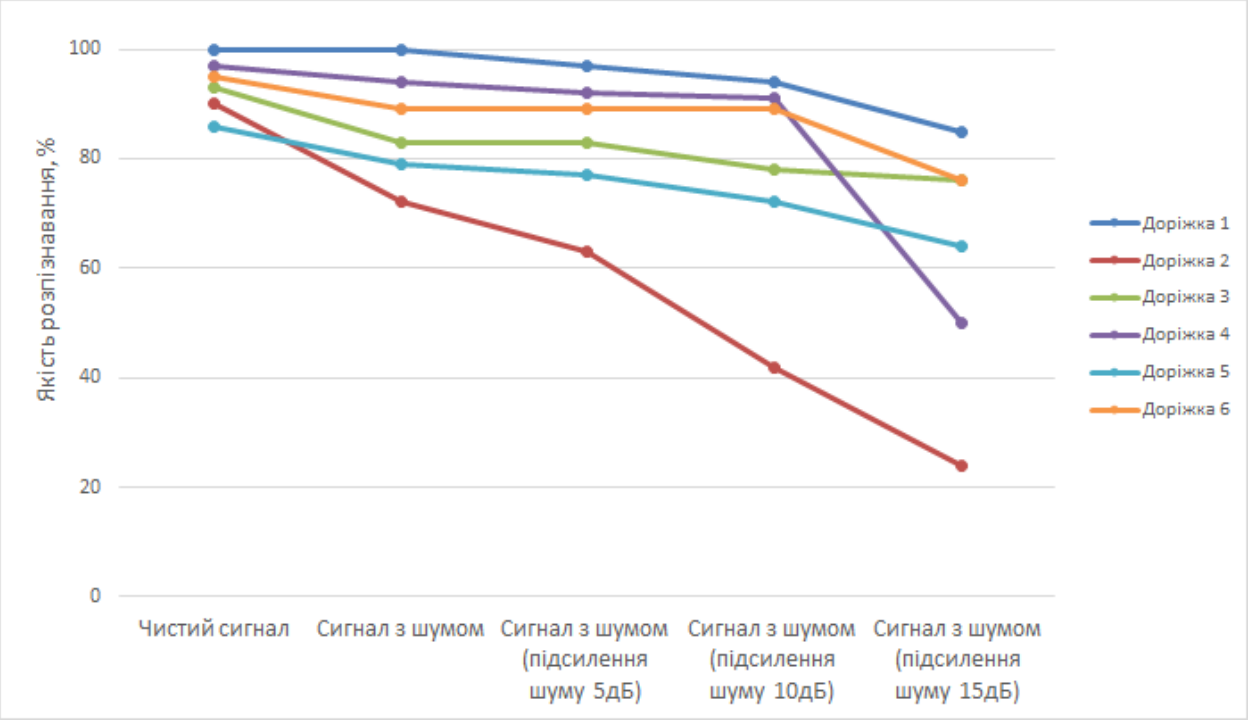


Рисунок 9. Результати розпізнавання сигналу з накладеним шумом ресторану

Як слідує з рис. 9, з підвищенням підсилення шуму, якість обробки

мови значно падає, у випадку звукової доріжки 2, майже унеможливлює користування програмою для розпізнавання мови при підсиленні 10 і більше дБ. Також з графіку видно, що відношення сигнал/шум звичайно впливає, але не є вирішальним фактором в чистоті розпізнавання мови. Чіткість вимови та наявність часто вживаних слів в звуковій доріжці є більш визначальними при розрахунку якості розпізнавання мови.

Проаналізуємо отримані дані при накладанні шуму залізничної станції за тим же самим принципом, як в попередньому випадку. Результати занесемо до табл. 2.

Таблиця 2. Розпізнавання сигналу з накладеним шумом залізничної станції

№	Чистий сигнал, %	Сигнал з шумом, %	Сигнал з шумом (підсилення шуму 5дБ) , %	Сигнал з шумом (підсилення шуму 10дБ) , %	Сигнал з шумом (підсилення шуму 15дБ) , %	Відношення сигнал/шум
1.	100	100	97	94	88	20.9595
2.	90	72	64	53	28	23.1170
3.	93	83	86	83	78	22.2780
4.	97	94	92	81	57	17.2141
5.	86	79	78	76	72	16.6584
6.	95	89	89	85	81	21.6558

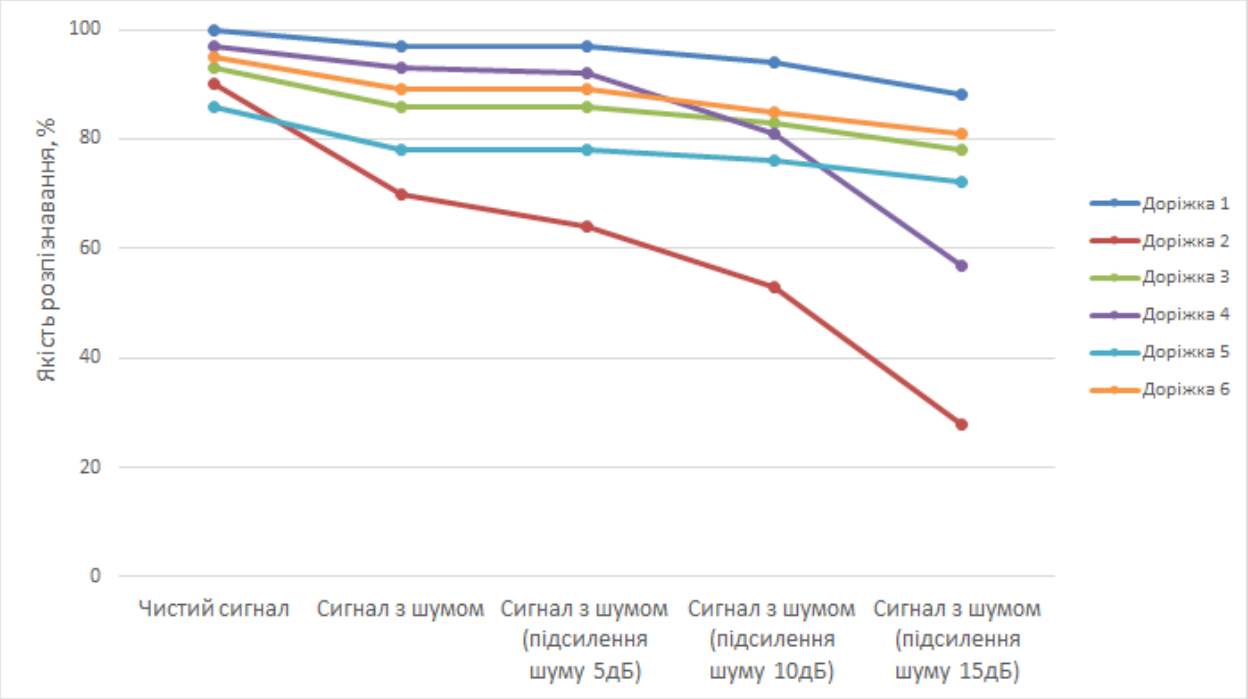


Рисунок 10. Результати розпізнавання сигналу з накладеним шумом залізничної станції

Змінюючи шум та відповідно відношення сигнал/шум, можемо простежити вплив цього параметру на чистоту розпізнавання мови. Дані з рис. 10 підтверджують висновок, зроблений до ситуації з накладеним шумом ресторану.

4.3. Розпізнавання звукових доріжок з накладеною реверберацією

Для продовження дослідження інструменту для розпізнавання мови, створюємо ще одні поширені у житті умови для обробки сигналу. Накладаємо реверберацію тривалістю 0,29; 0,6; 0,89; 1,1; 1,4; 2,0; 2,7 с на обрані сигнали.

Для того, щоб отримати сигнал з заданою реверберацією, у середовищі Matlab реалізуємо згортку двох сигналів – чистим мовним сигналом та імпульсною характеристикою приміщень з відповідною тривалістю реверберації, за допомогою функції y = fftfilt(b, x).

Отримані результати обробки мови заносимо до табл. 3.

Таблиця 3. Розпізнавання сигналу з накладеною реверберацією

№	Чистий сигнал, %	$T_р$=0,29 с, %	$T_р$=0,6 с, %	$T_р$=0,89 с, %	$T_р$=1,1 с, %	$T_р$=1,4 с, %	Тр=2,0 с, %	$T_р$=2,7 с,%
1.	100	100	100	88	73	38	8	0
2.	90	34	16	0	0	0	0	0
3.	93	90	74	42	38	0	0	0
4.	97	85	85	34	26	0	0	0
5.	86	80	65	32	25	2	0	0
6.	95	79	68	26	15	0	0	0

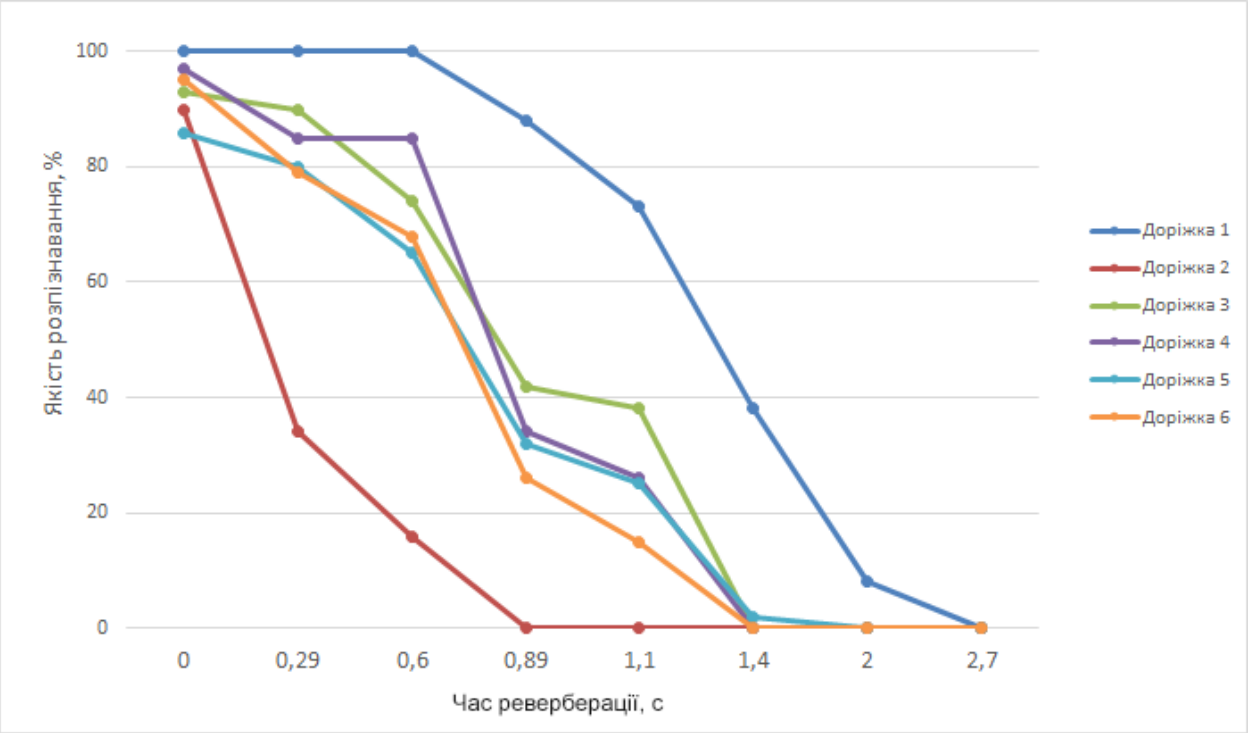


Рисунок 11. Результати розпізнавання сигналу з накладеною реверберацією

Як видно з рис. 11, розпізнавання мови стає незадовільної якості (приблизно 40 % і нижче) у більшості випадках починаючи з реверберації тривалістю 0,89 с. Простежується закономірність у спаданні якості розпізнавання мови у порівнянні з даними в підрозділі 4.2. Це говорить про те, що шуми і перешкоди, які виникають при неякісному записі мови, а також нечітка вимова слів відіграють визначну роль при розпізнаванні мови.

5. ПРАКТИЧНА РЕАЛІЗАЦІЯ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ АВТОМАТИЧНОГО РОЗПІЗНАВАННЯ МОВИ

Дуже часто при використанні системи розпізнавання мови мовні сигнали сприймаються за допомогою мікрофонів побутового рівня, в несприятливих для запису мови умовах. В той же час, комерційні системи розпізнавання мови, не передбачаючі додаткового навчання безпосередньо користувачем, використовують в основі свого мовного корпусу записи, вироблені, як правило, на високоякісній апаратурі и у відповідних умовах. В силу розбіжностей технічних і просторових каналів відбувається погіршення якості розпізнавання мови. Так як виробляти створення мовного корпусу, враховуючи велику кількість можливих умов запису, представляється економічно (великі витрати людських годин) і технічно (значно зросте займаний мовним корпусом простір на накопичувачах) недоцільним, логічною є необхідність у «вирівнюванні» каналів, через які проходит мова.

Найпростіший тракт запису, який широко використовується в побуті користувачами персональних комп’ютерів, представлений на рис. 92.

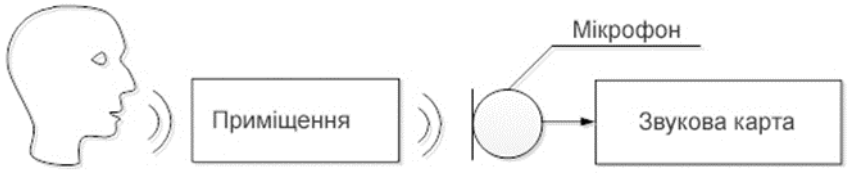


Рисунок 92. Тракт запису мовного сигналу

5.1. Нормалізація АЧХ входного тракту

На сьогоднішній день, задача компенсації відмінностей властивостей і характеристик різних мікрофонів, освітлена доволі слабо. В той же час, спеціалістами в області обробки кольорових зображень аналогічна задача

була вирішена застосуванням «кольорових профілей» приладів (таких як фотокамери, принтери, сканери, монітори та інші). «Кольоровий профіль» для випадку, наприклад, сканера — це калібрувальна таблиця, дозволяюча отримати ввідомості про різницю між кольоровим простором даного сканера і деяким другим кольоровим простором-еталоном. Використовуючи ці дані, можна програмним шляхом компенсувати нерівності в кольоровій чутливості сканера. В той же час, відомі методи створення «кольорових профілей» часто потребують наявності спеціальних умов освітлення і використання дорогого колориметра, або ж надзвичайно часозатратні і не гарантують коректності отриманих результатів. [7]

Найближчим аналогом сканера в області апаратури для роботи зі звуком є мікрофон. Провівши аналогії з процесом створення «кольорових профілей», в даному підрозділі запропонований метод отримання «звукового профіля» мікрофона. Метод засновується на вивчені АЧХ мікрофона, для якого складається «звуковий профіль». Як наслідок, для його створення необхідно визначити АЧХ мікрофона якомога більш точним чином. Найбільш підходящими методами в контексті цілі дослідження представляють метод безпосереднього виміру і метод порівняння.

5.1.1. Метод безпосереднього виміру

При отриманні АЧХ мікрофона даним методом передбачається, що запис відбувається в умовах вільного поля (наприклад, в безеховій камері), а АЧХ випромінювача рівномірна на всьому діапазоні сприймаємих людиною частот. Випромінюваний сигнал представляє собою реалізацію білого шуму тривалістю не менше 30 с. Рівень вихідного сигналу випромінювача встановлюється таким чином, щоб на виході мікрофону була відсутня перегрузка по рівню. Досліджуваний мікрофон встановлюється на відстані 1 м від випромінювача так, щоб акустичні вісі випромінювача і мікрофона

були перпендикулярні один одному.

Безпосередньо АЧХ мікрофона можна отримати, взявши модуль перетворення Фур'є записаного шуму

$$A_{мик} = \left|H_{мик}(f)\right| = \left|\mathcal{F}\{Y(t)\}\right|, \qquad (2.54)$$

де $\mathcal{F}$ — дискретне перетворення Фур'є, $Y(t)$ — сигнал на выході мікрофону.

Даний метод є найбільш точним методом отримання АЧХ і по можливості має використовуватись завжди, якщо є можливість забезпечити необхідні умови запису. Для запису в домашніх умовах допустимим вважається використовувати приміщення з невеликою кількістю відзеркалюючих поверхонь, а в якості випромінювача для відворення білого шуму — якісну широкополосну акустичну систему.

5.1.2. Метод порівняння

Метод порівняння є відносним методом оцінки АЧХ мікрофона. В реальних експериментах за допомогою даного методу можна отримати непогані результати, хоч і менш точні в порівнянні з методом безпосереднього виміру.

Суть методу заключається в використанні додаткового еталоного мікрофона з завчасно відомою АЧХ.

При використанні метода порівняння не висуваються настільки ж сурові вимоги до умов запису і використаній аппаратурі. Використовуючи другий мікрофон, простим відніманням спектрів можна з'ясувати, яких змін зазнає спектр в порівнянні з еталоним для цього мікрофону АЧХ, отриманим виробником методом безпосереднього виміру, або будь-яким іншим. Таким чином, вираз для АЧХ досліджуваного мікрофона можна записати як

$$A_{мик} = \left|H_{мик}(f)\right| = \left|\mathcal{F}\{Y(t)\}\right| + \left(A_{эт.АЧХ} - \left|\mathcal{F}\{Z(t)\}\right|\right), \qquad (2.55)$$

де $Y(t)$ — сигнал на виході досліджуваного мікрофона; $A_{эт.АЧХ}$ — априорно

відома АЧХ еталоного мікрофона; $Z(t)$— сигнал на виході еталоного мікрофона.

5.2. Усунення клацань і піків в сигналі

В ході експериментів, проведених в розділі 4, було виявлено, що наявність в мовному сигналі коротких «клацань» і звукових «сплесків», що виникають при відкритті рота, призводить до значного підвищення кількості помилкових спрацьовувань системи.

Був розроблений фільтр, дозволяючий ефективно усувати подібні артефакти в сигналі. Принцип роботы фільтра можна описати наступним алгоритмом.

1. Позиція початку фрейма $t_н$ встановлюється в нульовий момент часу.
2. З сигналу виділяється фрейм $F_{щел}$, який гіпотетично містить «клацання». Розраховується його питома енергія $E_{щел}$.

$$F_{щел} = [t_н; t_н + \tau_ф], \tag{2.56}$$

де $\tau_ф$ — передбачувана тривалість клацання.

Вираз для розрахунку енергії відрізка дискретного сигналу має вигляд

$$E = \sum_{n_1}^{n_2} x_i^2, \tag{2.57}$$

де n_1, n_2 — номери відліків початку і кінця фрейма, x_i — i-тий відлік сигналу.

3. Виділяються фрейми зліва і справа від фрейма $F_{щел}$. Позначимо їх $F_{лев}$ і $F_{прав}$ відповідно.

$$\begin{aligned} F_{лев} &= [t_н - \tau_{ан}; t_н]; \\ F_{прав} &= [t_н + \tau_ф; t_н + \tau_ф + \tau_{ан}], \end{aligned} \tag{2.58}$$

де $\tau_{ан}$ — тривалість фрейма аналізу.

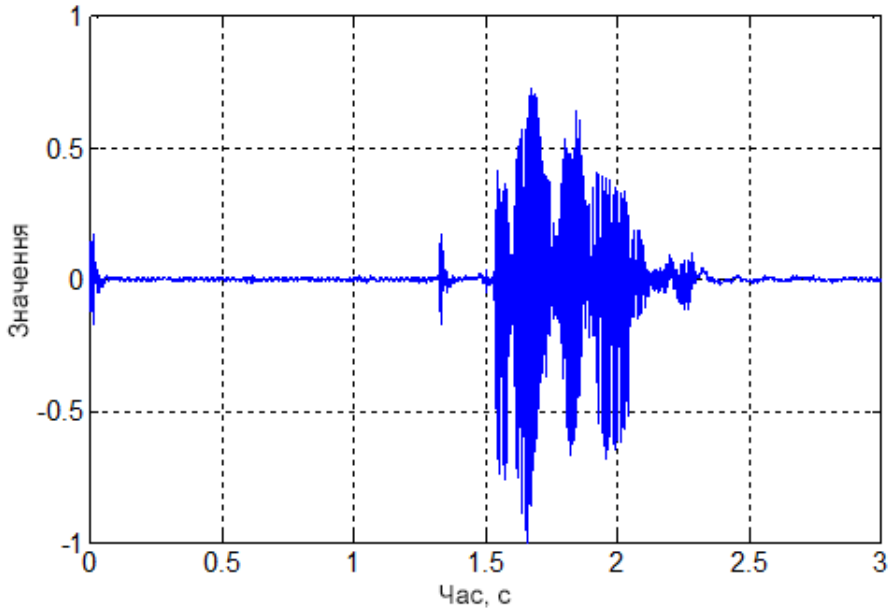
Прийняття позитивного рішення про наявність клацання в фреймі $F_{щел}$ приймається у випадку істинності рівності

$$E_{щел} > (E_{лев} + E_{прав}) \cdot k, \tag{2.59}$$

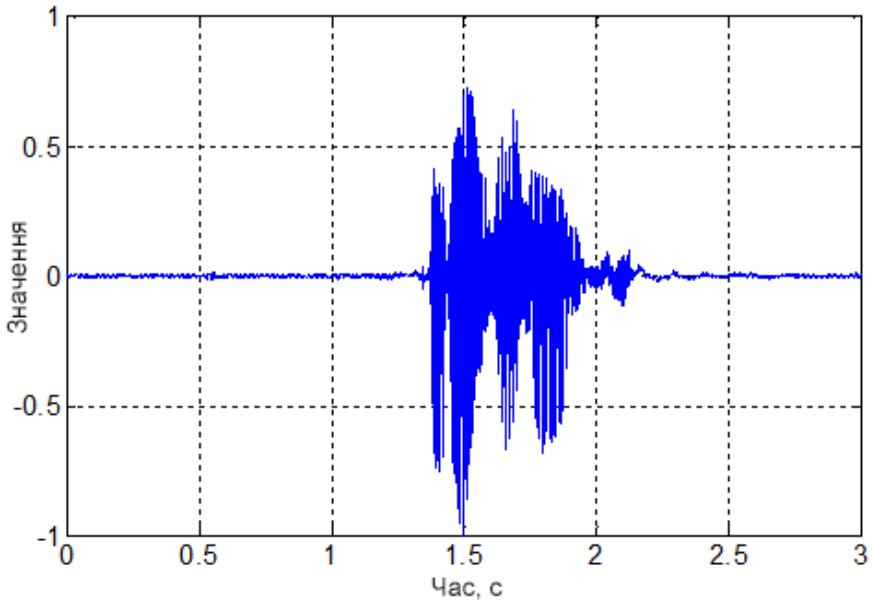
де k – деякий коефіцієнт, який залежить від $\tau_{ф}$ і $\tau_{ан}$.

4. Позиція початку фрейма $t_{н}$ здвигається вперед на час $\tau_{н}$.
5. Повторяти обробку з п. 2 до тих пір, поки не буде досягнутий кінець сигналу.

Приклад результатів обробки приведений на рис. 104.



а)



б)

Рисунок 104. Результат роботи детектора «клацань»

В даному випадку виявились усунуті два сплески: на початку в в кінці запису, в той час як інша частина сигналу залишилась неторкнутою.

5.1.3. Отримання звукового профіля мікрофону

Незалежно від методу отримання АЧХ микрофону, вираз для розрахунку «звукового профіля» досліджуваного мікрофона можна записати як

$$K_{зв.проф.} = \frac{A_{бел.шум}}{A_{мик}}, \tag{2.60}$$

де $A_{бел.шум}$ — амплітудний спектр білого шуму. Однак, так спектр білого шуму в идеалі представляє собою константу, вираз (2.60) можна записати як

$$K_{зв.проф.} = \frac{1}{A_{мик0}}, \tag{2.61}$$

де $A_{мик0}$ — приведена до одиниці АЧХ мікрофона, яку можна отримати за допомогою виразу

$$A_{мик0} = \frac{A_{мик}}{\max(A_{мик})}, \tag{2.62}$$

де $\max(A_{мик})$ — амплітудне значення АЧХ мікрофона.

Приклад того, як може виглядати такий «звуковий профіль», приведений на рис. 113. В якості досліджуваного мікрофона використовувався мікрофон мобільного телефону Motorola Defy.

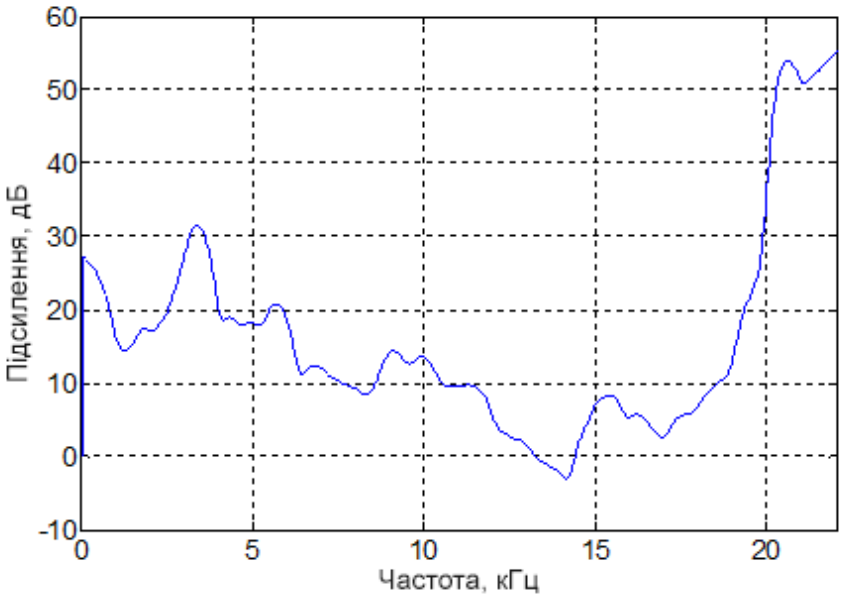


Рисунок 113. Приклад «звукового профіля»

Як видно з рис. 113, даний мікрофон демонструє значну нерівномірність АЧХ на всьому спектрі частот.

Спосіб застосування «звукового профіля» до записанного сигналу описується наступним виразом:

$$N_{корр}(t) = \mathcal{F}^{-1}\left\{\mathcal{F}\left\{X(t)\right\}\cdot K_{зв.проф}\right\}. \qquad (2.63)$$

де $N_{корр}(t)$— сигнал, до якого застосований «звуковий профіль»; $X(t)$ — сигнал, підвержений корекції.

ВИСНОВОК

В роботі описані дослідження можливих шляхів підвищення ефективності автоматичного розпізнавання мови.

Проведено огляд поточного стану досліджень в напрямку розпізнавання мови. З достатнім рівнем впевненості можна стверджувати, що якість автоматичного розпізнавання мови з часом зростатиме. В цілому, ймовірно, підвищити надійність розпізнавання до рівня розпізнавання мови людиною в найближчому майбутньому не вдасться.

Описано основні елементи теорії систем розпізнавання мови, побудованих на основі прихованих марківських моделей. Наведено загальні відомості про марковскі процеси, а також про спосіб використання ПММ в якості параметричної моделі мовного сигналу. Розглянуто основні ідеї оцінки параметрів ПММ: рекурентне оцінювання Баума-Уелча і алгоритм Вітербі. Описано найбільш ефективні методи параметризації мовних сигналів: лінійне передбачення і використання мел-кепстральних коефіцієнтів.

Проведено огляд основних можливостей і функціонала програмного інструментарію HTK, описано застосування його для досліджень в напрямку розпізнавання мови. Наведено загальну схему і принципи роботи з HTK, зазначено призначення основних бібліотечних модулів пакета.

Розглянуто можливості застосування мови програмування Python у вирішенні задачі розпізнавання мови. Пояснено принцип роботи бібліотеки SpeechRecognition, можливості якої і використовувалися при проведенні експериментальної частини роботи.

Проведені експериментальні дослідження залежності якості автоматичного розпізнання мови від таких факторів, як наявність фонового шуму та реверберації. Дослідження показали критичний рівень фонових шумів та тривалості реверберації для розпізнавання усної мови.

Проведена підготовча частина роботи по розробці детектора сторонніх звуків і шумів, що створюються мовним апаратом людини, а також деяких

видів мовних збоїв, що виникають у спонтанній вимові.

Запропоновано метод отримання «звукового профілю» мікрофона, за допомогою якого можливо «вирівняти» АЧХ сигналу, записаного на цей мікрофон, з метою поліпшення якості звучання. Описано два методи отримання АЧХ мікрофона, між якими можна здійснювати вибір в залежності від доступних умов запису та апаратури. Однак, забезпечити наявність такої апаратури у користувача системи автоматичного розпізнавання мови не є можливим. У зв'язку з цим ймовірним напрямком роботи є розробка методу «сліпої» приблизної оцінки АЧХ мікрофона. В цілому можливість практичного використання «звукових профілів» мікрофонів є важливим питанням, гідним подальшого розвитку не тільки в даній роботі, а й в цілому.

В роботі залишлося нерозглянуте важливе питання практичної реалізації придушення фонових шумів і звуків. Дана проблема ускладнюється тим, що в ряді випадків немає можливості вдатися до адаптивної фільтрації, так як запис чистого шуму, вироблена одночасно із записом мовного сигналу, часто не існує. В силу цього необхідне подальше вивчення можливості застосування сліпих адаптивних фільтрів для даної задачі.

Одним з важливих напрямків подальшої роботи також є розробка детектора вокалізованих пауз, як частини комплексу фільтрів, спрямованих на поліпшення якості мовних сигналів.

Побудова системи автоматичного розпізнавання мови є актуальним завданням на сьогоднішній день, і не дивлячись на наявність великої кількості об'єктивних труднощів і невирішених проблем, задіяні алгоритми і принципи будуть продовжувати вдосконалюватися безліччю компаній і незалежних дослідників.

СПИСОК ЛІТЕРАТУРИ

1. Математические методы решения военно-специальных задач / В. З. Казачинский, Г.Е. Левитский. — Москва, 1980.

2. Скрытая марковская модель — Википедия. [Електронний ресурс] // wikipedia.org – Режим доступу до ресурсу: http://ru.wikipedia.org/wiki/Скрытая_марковская_модель

3. Дідковський В. С., Продеус А. М., «Дослідження шляхів підвищення ефективності комп'ютерних систем слухомовної корекції людей з порушеннями слуху та глухотою». НТУУ «КПІ», Київ, 2008.

4. Сычёв А. В. Скрытые марковские модели [Електронний ресурс] // ru.wikibooks.org – 2012 – Режим доступу до ресурсу: http://ru.wikibooks.org/wiki/Скрытые_марковские_модели

5. «The HTK Book» [Електронний ресурс] // htk.eng.cam.ac.uk – Режим доступу до ресурсу: http://htk.eng.cam.ac.uk/docs/docs.shtml

6. Урицкий И. Хабрахабр. [Електронний ресурс] // habrahabr.ru – Режим доступу до ресурсу: http://habrahabr.ru/post/140828

7. ICC profile [Електронний ресурс] // wikipedia.org – Режим доступу до ресурсу: http://en.wikipedia.org/wiki/ICC_profile

8. The Python Package Index (PyPI) [Електронний ресурс] // pypi.org – Режим доступу до ресурсу: https://pypi.org/

9. Google cloud Speech-to-Text [Електронний ресурс] // cloud.google.com – Режим доступу до ресурсу: https://cloud.google.com/speech-to-text/

10. SpeechRecognition library [Електронний ресурс] // pypi.org – Режим доступу до ресурсу: https://pypi.org/project/SpeechRecognition/

11. The Ultimate Guide To Speech Recognition With Python [Електронний ресурс] // realpython.com – Режим доступу до ресурсу: https://realpython.com/python-speech-recognition/

ДОДАТОК

Код програми для розпізнавання мови.

```
import speech_recognition as sr

from os import path

from pprint import pprint

audio_file = path.join(path.dirname(path.realpath(__file__)),
"YourFile.wav")

r = sr.Recognizer()
with sr.AudioFile(audio_file) as source:
    audio = r.record(source)

try:
    text = r.recognize_google(audio,show_all=True)
    pprint (text)
except:
    print ("Didn't work.")
```